Resolving Degeneracy

in Combinatorial Linear Programs:

Steepest Edge, Steepest Ascent, and Closest Ascent

E. Andrew Boyd[1]

July, 1991

TR91-21

Report Documentation Page

1. REPORT DATE **JUL 1991**	2. REPORT TYPE	3. DATES COVERED **00-00-1991 to 00-00-1991**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Resolving Degeneracy in Combinatorial Linear Programs: Steepest Edge, Steepest Ascent, and Closest Ascent	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Computational and Applied Mathematics Department ,Rice University,6100 Main Street MS 134,Houston,TX,77005-1892**	8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**27**	

Abstract

While variants of the steepest edge pivoting rule are commonly used in linear programming codes they are not known to have the theoretically attractive property of avoiding an infinite sequence of pivots at points of degeneracy. A natural extension of the steepest edge pivoting rule based on steepest ascent is developed and shown to be provably finite. An alternative finite pivoting procedure that is computationally more attractive than steepest ascent is then introduced and it is argued that with probability 1 the procedure has the same computational requirements as steepest edge *independent* of the linear program being solved. Both procedures have the unique advantage that they choose the pivot element without explicit knowledge of the set of all active constraints at a point of degeneracy, thus making them attractive in combinatorial settings where the linear program is never explicitly written out.

1 Introduction

Degeneracy in simplex algorithms for solving linear programs has long been recognized as both a theoretical and practical problem. From a theoretical perspective the problem was resolved early in the history of linear programming. The perturbation/lexicographic method (Charnes, 1953; Dantzig, Orden, and Wolfe, 1955; Wolfe, 1963) avoids cycling by dictating the leaving variable while allowing complete freedom in the choice of entering variable (among variables with a reduced cost of correct sign). Bland's combinatorial rule (1977) dictates the choice of both entering and leaving variable. More recently, Magnanti and Orlin (1988) proposed a rule for the parametric simplex algorithm that avoids degeneracy using a lexicographic method that dictates the entering variable while maintaining complete freedom in the choice of leaving variable (among variables that would drive the solution infeasible).

The practical difficulties associated with degeneracy have come to be fully appreciated far more slowly. To this day many texts on linear programming, having resolved the problem of degeneracy theoretically, routinely pronounce that degeneracy is not a practical problem based on the observation that cycling rarely occurs. Yet, even if cycling does not occur, it is nonetheless possible to encounter a very long sequence of degenerate pivots. Cunningham (1979) discussed the possibility of encountering an exponential number of such pivots without cycling in the context of the network simplex algorithm and introduced the term stalling for this phenomenon. While all linear programs arising in practice are degenerate to some degree, the number of degenerate pivots encountered by a simplex algorithm is profoundly influenced by the pivot selection rule. Ryan and Osborne (1988) reported that due to degeneracy they were unable to solve an airline crew scheduling problem after approximately

1

2000 iterations when using a pivot rule based solely on reduced cost information, but that they solved the problem in 436 iterations using Wolfe's rule (1963). The author (Boyd, 1990) was able to solve some extremely degenerate combinatorial linear programs only after employing the steepest edge pivot rule. For linear programs demonstrating significant degeneracy — such as linear programs arising in combinatorial contexts — the choice of pivot rule can mean the difference between solving or not solving a linear program.

One pivot rule of particular importance is steepest edge. The steepest edge pivot rule consists of choosing the edge of ascent relative to a given basis that gives the greatest change in the objective function per unit of movement along that edge. Steepest edge and its variants have been known and used for decades in the solution of large-scale linear programs and many papers have been written on the subject, including Crowder and Hattingh (1974), Dickson and Frederick (1960), Goldfarb and Reid (1977), Harris (1973), Kuhn and Quandt (1963), and Wolfe and Cutler (1963). The major conclusion reached by these studies is that steepest edge often dramatically reduces the number of pivots required by the simplex algorithm. Recent extensive computational results supporting this claim have been obtained by Bixby (1991) and by Forrest and Goldfarb (1991). In spite of the additional computational work required at each iteration over simpler pivot rules, steepest edge often proves to be the pivot rule of choice due to the increased speed attained by a smaller number of iterations. Hoffman and Padberg (1991) have reported marked improvement in the solution times of airline crew scheduling problems with the use of steepest edge. Commercial codes such as IBM's MPSX and OSL both have options for DEVEX pricing (an approximate steepest edge procedure) and CPLEX makes use of primal and dual steepest edge pricing.

Beyond the overall effectiveness of steepest edge, a common belief within the linear

2

programming community is that steepest edge generates relatively few pivots at points of degeneracy. The author became interested in the steepest edge pivot rule when it was suggested as a way to overcome degeneracy encountered in his own applications. However, while steepest edge works well in practice it remains an open question as to whether or not steepest edge is a true degeneracy resolution procedure; that is, a procedure that is guaranteed to determine an edge of ascent or a proof that no such edge exists after a finite number of pivots.

In this paper we propose two simplex pivot rules closely related to steepest edge that are degeneracy resolution procedures. The first is a very natural extension of the steepest edge rule based on steepest ascent. The second procedure is computationally far more attractive than steepest ascent and it is argued that with probability 1 the procedure has the same computational requirements as steepest edge *independent* of the linear program being solved.

The procedures are different than all previous degeneracy resolution procedures in that they rely neither on lexicography nor ordering arguments such as those found in Bland's rule. They share some simlilarity with the procedure of Magnanti and Orlin in that there are no restrictions placed on the leaving variable, and the steepest ascent procedure actually provides some freedom in the choice of entering variable. However, the most important theoretical advantage unique to the procedures is that the pivot element is chosen without explicit knowledge of the set of all active constraints at a point of degeneracy. This advantage is particularly useful in combinatorial settings where the linear program is never explicitly written out.

2 Background

In order to make the combinatorial implications of the proposed degeneracy resolution procedures explicit, we consider a linear program of the form

$$\max \quad cx$$

$$(P) \quad \text{s.t.} \quad Ax \leq b$$

where A is $m \times n$ and is assumed to have rank n (the problem is, of course, trivial to solve if the rank is less than n). It is important to emphasize that while the form (P) will be used for expository purposes the procedures to be described can be applied just as easily to problems in a more standard form for simplex algorithms, i.e., equality constraints and bounded variables. In the context of (P), a primal simplex algorithm operates as follows.

Algorithm Simplex

Given: A linearly independent collection $a^1, \ldots, a^n$ of rows of A such that the intersection $\bar{x}$ of the corresponding constraints is feasible for (P).

Purpose: Solve (P).

1. For each of the n sets of points $S_k = \{a^1, \ldots, a^n\} - \{a^k\}$ let $\gamma^k a \leq 0$ be the unique direction satisfying $\gamma^k a^i = 0$ for all $a^i \in S_k$ and $\gamma^k a^k < 0$. Let $d = \gamma^s$ where γ^s satisfies $c\gamma^s > 0$. If no such direction exists, stop; the present solution $\bar{x}$ is optimal.

2. Find the largest $\hat{\theta}$ such that $A(\bar{x} + \hat{\theta}d) \leq b$ and a row r of A such that $a^r(\bar{x} + \hat{\theta}d) > b_r$ for $\theta > \hat{\theta}$. If $\hat{\theta} = \infty$, stop; (P) is unbounded.

3. Let the new set of constraints consist of $\{a^1, \ldots, a^n\} \cup \{a^r\} - \{a^s\}$ and return to step 1.

In the standard form for simplex algorithms the leaving constraint a^s would correspond to an entering variable and the entering constraint a^r would correspond to a leaving variable. An important point can be made with respect to step 2 of algorithm Simplex. When the constraints $Ax \le b$ are explicitly written out step 2 can be carried out enumeratively; that is, for each constraint $a^i x \le b_i$ the value θ_i can be determined such that $a^i(\bar{x} + \theta_i d) = b_i$ and $\hat{\theta}$, a^r, and b_r determined by the minimum non-negative value of θ_i. In standard implementations of the simplex algorithm this is what actually occurs. However, to actually apply algorithm Simplex all that is necessary is an oracle for solving the following problem.

Problem STEP: Given a point $\bar{x}$ satisfying $A\bar{x} \le b$ and a direction d, find the largest $\hat{\theta}$ such that $A(\bar{x} + \hat{\theta}d) \le b$ and a row r of A such that $a^r(\bar{x} + \hat{\theta}d) > b_r$ for $\theta > \hat{\theta}$.

For linear programs (P) arising in combinatorial contexts it is often the case that $m >> n$ so that explicitly writing the set of constraints is neither desirable nor practically feasible. However, in many such instances an oracle for solving problem STEP does exist. Motivation for the present work stemmed from degeneracy encountered in the solution of knapsack separation problems (Boyd, 1990) where the linear programs that arise have an exponential number of constraints but an efficient oracle for problem STEP. Algorithms for finding violated subtour elimination constraints of the traveling salesman polytope can be modified to serve as oracles for problem STEP, allowing the simplex algorithm to be used to optimize over the subtour polytope as defined in Boyd and Pulleyblank (1990) in spite of an exponential number of constraints. The problem of finding a direction of ascent on the cone defined by simple cycle inequalities arises in the solution of the maximum weight cut problem (Barahona and Titan, 1990; Bixby and Saigal, 1991), and again while there are an exponential number of such constraints there exists an efficient oracle for problem STEP.

Yet, while oracles often exist for problem STEP, most degeneracy resolution procedures must place requirements on the entering constraint (exiting variable in standard simplex) in order to guarantee resolution of degeneracy. A notable exception is the parametric algorithm of Magnanti and Orlin (1988), although as a parametric algorithm it requires more than an oracle for problem STEP in order to be implemented.

In addition, all degeneracy resolution procedures proposed to date require global information on the set of active constraints at a point of degeneracy in order to make a pivot selection. To elaborate, algorithm Simplex operates by successively choosing not only a feasible point $\bar{x}$ but a set of n constraints with linearly independent gradients. In the context of degenerate pivots, it is the choice of constraints that is particularly relevant since by definition the feasible point remains unchanged. Bland's rule and rules that use lexicography require information on more than the chosen set of active constraints to make a pivot selection. However, the procedures described here require only knowledge of the chosen set of n constraints at any given iteration to choose a pivot.

Throughout this paper we let $\| \cdot \|$ denote the euclidean norm. We use $cone(a^1, \ldots, a^n)$ to denote $\{a \in \mathbb{R}^n : a = \omega_1 a^1 + \ldots + \omega_n a^n, \ \omega_1, \ldots, \omega_n \geq 0\}$.

3 Steepest Edge

The steepest edge pivot rule defines the way in which the direction d is chosen in step 1 of algorithm Simplex. Specifically, d is chosen as the direction γ^s with the largest increase in objective function value per unit of movement along γ^s; that is, the steepest edge of the polyhedron defined by the chosen collection $a^1, \ldots, a^n$ of active constraints. We formally

define the algorithm in the context of (P) for comparison with algorithms presented in the following sections.

Algorithm Steep

Given: A linearly independent collection $a^1, \ldots, a^n$ of rows of A such that the intersection $\bar{x}$ of the corresponding constraints is feasible for (P).

Purpose: Determine a direction of ascent at $\bar{x}$ or prove that no such direction exists.

1. For each of the n sets of points $S_k = \{a^1, \ldots, a^n\} - \{a^k\}$ let $\gamma^k a \leq 0$ be the unique direction satisfying $\gamma^k a^i = 0$ for all $a^i \in S_k$ and $\gamma^k a^k < 0$.

2. Let γ^s maximize $v = (\gamma^i c)^2 / \|\gamma^i\|^2$ for all $\gamma^1, \ldots, \gamma^n$. If $v \leq 0$, stop; no ascent direction exists.

3. Let $d = \gamma^s$.

4. Let $\hat{\theta}$ and a^r be the values returned by the oracle for problem STEP with d as defined in step 3. If $\hat{\theta} \neq 0$, stop; d is an ascent direction.

5. Replace a^s with a^r and return to step 1.

4 Steepest Ascent

While the status of steepest edge as a theoretical procedure for resolving degeneracy remains unresolved, it turns out that a close variant of this algorithm based on steepest ascent is provably a degeneracy resolution procedure. Specifically, if the direction d sent to the oracle for solving problem STEP is a direction of steepest ascent relative to the active constraints

(as opposed to a steepest edge) then algorithm Simplex cannot cycle. Formally, we introduce the following algorithm which, given a vertex of the feasible region of (P), finds a direction of ascent or proves that no such direction exists after a finite number of iterations.

Algorithm Ascent

Given: A linearly independent collection $a^1, \ldots, a^n$ of rows of A such that the intersection $\bar{x}$ of the corresponding constraints is feasible for (P).

Purpose: Determine a direction of ascent at $\bar{x}$ or prove that no such direction exists.

1. Find a^0 such that $\|c - a^0\| \leq \|c - a\|$ for all $a \in cone(a^1, \ldots, a^n)$ and let $\omega_1, \ldots, \omega_n \geq 0$ be the unique values such that $a^0 = \omega_1 a^1 + \cdots + \omega_n a^n$. If $a^0 = c$, stop; no ascent direction exists. Otherwise, let I_0 be the nonempty set of indices for which $\omega_i = 0$.

2. Let $\hat{\theta}$ and a^r be the values returned by the oracle for problem STEP with $d = c - a^0$. If $\hat{\theta} \neq 0$, stop; $c - a^0$ is an ascent direction.

3. Let a^s be such that $s \in I_0$ and $\{a^1, \ldots, a^n\} \cup \{a^r\} - \{a^s\}$ is linearly independent. Replace a^i with a^r and return to step 1.

Algorithm Ascent is a steepest ascent algorithm because $c - a^0$ is the direction of steepest ascent relative to c and the active constraint set defined by $a^1, \ldots, a^n$. This follows from strong duality results on separation found in the convexity theory literature. The following proofs actually require only the weaker results captured in the following lemma.

Lemma 1 *Let $a^1, \ldots, a^n \in \mathbb{R}^n$ be a collection of linearly independent points, let $c \in \mathbb{R}^n$ be such that $c \notin cone(a^1, \ldots, a^n)$, and let a^0 be such that $\|c - a^0\| \leq \|c - a\|$ for all*

$a \in cone(a^1, \ldots, a^n)$. *Finally, let* $\omega_1, \ldots, \omega_n \geq 0$ *be the unique values such that* $a^0 = \omega_1 a^1 + \cdots + \omega_n a^n$ *with* I_0 *the set of indices for which* $\omega_i = 0$ *and* I_0^C *the complement of* I_0. *Then*

1. $(c - a^0)a \leq 0$ *for all* $a \in cone(a^1, \ldots, a^n)$,

2. $(c - a^0)a^i = 0$ *for all* a^i *with* $i \in I_0^C$, *and*

3. $(c - a^0)c > 0$.

Geometrically, Lemma 1 is obvious and we omit the proof as it is easily established and one of many standard results in convexity theory.

Theorem 1 *Algorithm Ascent terminates after a finite number of iterations with a direction of ascent for (P) or a proof that no such direction exists.*

Proof. We begin by showing that when the algorithm terminates in step 1 no ascent direction exists and when it terminates in step 2 an ascent direction has been found. The constraints $a^1, \ldots, a^n$ are initially linearly independent and the algorithm proceeds beyond step 2 only if the value $\hat{\theta}$ returned by the oracle for solving problem STEP is 0. By the definition of problem STEP this means the constraint $a^r x \leq b_r$ returned by the oracle satisfies $a^r \bar{x} = b_r$, and so throughout the algorithm all constraints $a^i x \leq b_i$ that are encountered are satisfied at equality by $\bar{x}$. If the algorithm terminates in step 1 with $c = a^0$ then $c = \omega_1 a^1 + \cdots + \omega_n a^n$ with $\omega_i \geq 0$ and thus by Farkas' lemma no ascent direction in (P) exists at $\bar{x}$. If $c \neq a^0$ then $c - a^0$ is clearly a direction of increasing objective function value by Lemma 1, and if the value $\hat{\theta}$ returned by the oracle for solving problem STEP is

9

nonzero, as is the case when the algorithm terminates in step 2, then by the definition of problem STEP $c - a^0$ is a feasible direction for (P) at $\bar{x}$.

We next show that in step 1 if $c \neq a^0$ then $I_0 \neq \emptyset$. If $I_0 = \emptyset$ then a^0 resides in the strict interior of $cone(a^1, \ldots, a^n)$. By the operation of the algorithm $c \neq a^0$ and so the line segment connecting a^0 and c contains points other than c, and since $\| \cdot \|$ is a norm any point $a \neq a^0$ on this line segment satisfies $\|c - a\| < \|c - a^0\|$. However, since a^0 is contained in the strict interior of $cone(a^1, \ldots, a^n)$ some point $a \neq a^0$ on this line segment also must be contained in $cone(a^1, \ldots, a^n)$, contradicting the definition of a^0.

To see that a vector $a^s \in I_0$ with the properties assumed in step 3 exists, note that the algorithm only proceeds to step 3 if $\hat{\theta} = 0$, and by the definition of problem STEP it follows that a^r satisfies $(c - a^0)a^r > 0$. Since $(c - a^0)a^r \neq 0$ the set of vectors $\{a^i : i \in I_0^C\}$, which satisfy $(c - a^0)a^i = 0$ by Lemma 1, together with a^r are linearly independent. As the vectors $a^1, \ldots, a^n$ are linearly independent it is thus possible to find a vector a^s with $s \in I_0$ such that $\{a^1, \ldots, a^n\} \cup \{a^r\} - \{a^s\}$ is linearly independent.

We conclude by demonstrating that the algorithm is finite by showing that $\|c - a^0\|$ strictly decreases in each successive iteration of the algorithm. Since each possible set of constraints $a^1, \ldots, a^n$ defines a unique point a^0 (a^0 represents the minimum of a strictly convex function on a cone) no set $a^1, \ldots, a^n$ can ever be repeated and finite termination of the algorithm follows. Let $\bar{a}^1, \ldots, \bar{a}^n$ be the new set of vectors constructed in step 3 of the algorithm. The vector $a^r \in cone(\bar{a}^1, \ldots, \bar{a}^n)$ since $a^r = \bar{a}^k$ for some k by choice, and clearly $a^0 \in cone(\bar{a}^1, \ldots, \bar{a}^n)$ since a^0 can be written as a non-negative linear combination of the vectors $\{a^i : i \in I_0^C\}$ and $\{a^i : i \in I_0^C\} \subseteq \{\bar{a}^1, \ldots, \bar{a}^n\}$. The square of the distance between

c and points on the line segment $a^0 + \theta(a^r - a^0)$, $\theta \in [0,1]$, connecting a^0 and a^r is

$$\|c - (a^0 + \theta(a^r - a^0))\|^2$$

$$= ((c - a^0) - \theta(a^r - a^0))((c - a^0) - \theta(a^r - a^0))$$

$$= (c - a^0)(c - a^0) - 2\theta(c - a^0)(a^r - a^0) + \theta^2(a^r - a^0)(a^r - a^0)$$

$$= \|c - a^0\|^2 - 2\theta[(c - a^0)a^r - (c - a^0)a^0] + \theta^2\|a^r - a^0\|^2.$$

Since $(c - a^0)a^r > 0$ and $(c - a^0)a^0 = 0$ it follows that $\|c - (a^0 + \theta(a^r - a^0))\| < \|c - a^0\|$ for θ sufficiently small but positive. This in turn implies that the point $\overline{a}^0$ satisfying $\|c - \overline{a}^0\| \leq \|c - a\|$ for all $a^0 \in cone(\overline{a}^1, \ldots, \overline{a}^n)$ satisfies $\|c - \overline{a}^0\| < \|c - a^0\|$, completing the proof. $\quad\square$

It is possible that algorithm Ascent will terminate in step 2 with a direction of ascent that is not an edge. Practically this poses no difficulty as an extreme point of (P) with a better objective function value than $\bar{x}$ can be trivially constructed given an ascent direction at $\bar{x}$, and finding an edge of ascent at $\bar{x}$ is not significantly more difficult.

The choice of the constraint a^r returned by the oracle for solving problem STEP is irrelevant in guaranteeing the finiteness of algorithm Ascent. However, it will have an important effect on the speed with which degeneracy is resolved in practice. In particular, if an a^r is generated that greatly decreases the distance from c to the associated cone at each iteration then rapid resolution of degeneracy should follow.

While Theorem 1 does not prove that steepest edge is a degeneracy resolution procedure it does shed some light on why steepest edge is an effective degeneracy resolution procedure in practice. Often the closest point to c in $cone(a^1, \ldots, a^n)$ may well be a facet of this cone or, equivalently, the direction of steepest ascent may well be the steepest edge. Even if the direction of steepest ascent is not the steepest edge, the steepest edge may be a sufficiently good approximation to the direction of steepest ascent so that $\|c - a^0\|$ decreases nonetheless.

5 Closest Ascent

While steepest ascent is theoretically sound it relies upon finding a point a^0 minimizing $\|c - a^0\|$ on an n-dimensional cone with n extreme rays. While this problem can be solved finitely using an active set strategy it nonetheless remains a non-trivial subproblem that should be avoided if at all possible.

In this section we present a degeneracy resolution procedure that depends only upon the existence of an oracle for solving problem STEP but finds an a^0 minimizing $\|c - a^0\|$ only as a last resort. In fact, following the discussion of the algorithm it will be argued that the probability of having to solve such a subproblem is 0 independent of the given degenerate point.

Algorithm Close

Given: A linearly independent collection $a^1, \ldots, a^n$ of rows of A such that the intersection $\bar{x}$ of the corresponding constraints is feasible for (P).

Purpose: Determine a direction of ascent at $\bar{x}$ or prove that no such direction exists.

0. Let a^0 be any point in the interior of $cone(a^1, \ldots, a^n)$.

1. For each of the n sets of points $S_k = \{a^1, \ldots, a^n\} - \{a^k\}$ let $\gamma^k a \leq 0$ be the unique constraint satisfying $\gamma^k a^i = 0$ for all $a^i \in S_k$ and $\gamma^k a^k < 0$.

2. Let $\hat{\alpha} \geq 0$ be the maximum value of α such that $\gamma^k(a^0 + \alpha(c - a^0)) \leq 0$ for all $\gamma^1, \ldots, \gamma^n$ and let K be the set of indices k such that $\gamma^k(a^0 + \hat{\alpha}(c - a^0)) = 0$.

3. if $|K| = 1$

Let $I_0 = K$, let $d = \gamma^k$ where $k \in K$, and let $a^0 = a^0 + \hat{\alpha}(c - a^0)$. If $\hat{\alpha} \geq 1$, stop; no ascent direction exists.

else

Let a^0 be such that $\|c - a^0\| \leq \|c - a\|$ for all $a \in cone(a^1, \ldots, a^n)$ and let $\omega_1, \ldots, \omega_n \geq 0$ be the unique values such that $a^0 = \omega_1 a^1 + \cdots + \omega_n a^n$. If $a^0 = c$, stop; no ascent direction exists. Otherwise, let I_0 be the nonempty set of indices for which $\omega_i = 0$ and let $d = c - a^0$.

4. Let $\hat{\theta}$ and a^r be the values returned by the oracle for problem STEP with d as defined in step 3. If $\hat{\theta} \neq 0$, stop; d is an ascent direction.

5. Let a^s be such that $s \in I_0$ and $\{a^1, \ldots, a^n\} \cup \{a^r\} - \{a^s\}$ is linearly independent. Replace a^s with a^r and return to step 1.

The key difference between algorithm Ascent and algorithm Close lies in how a^0 is chosen from iteration to iteration. In algorithm Ascent strict monotonicity of $\|c - a^0\|$ is guaranteed by choosing a^0 as the closest point to c in $cone(a^1, \ldots, a^n)$. In algorithm Close strict monotonicity is maintained by first checking to determine if a there exists a simple pivot that will yield a new value $\overline{a}^0$ which is closer to c on the line segment connecting a^0 and c at the next iteration. Only if such a pivot does not exist is $\overline{a}^0$ chosen as the closest point to c in $cone(a^1, \ldots, a^n)$. Computationally, this latter option is far more attractive.

Theorem 2 *Algorithm Close terminates after a finite number of iterations with a direction of ascent for (P) or a proof that no such direction exists.*

13

Proof. For notational convenience we let $\overline{a}^0$ be the new value of a^0 determined in step 3 of the algorithm and let $\overline{a}^1, \ldots, \overline{a}^n$ be the new values of $a^1, \ldots, a^n$ determined in step 5 of the algorithm.

We begin by showing that after each iteration of the algorithm $\overline{a}^0 \in cone(\overline{a}^1, \ldots, \overline{a}^n)$. Initially, $a^0 \in cone(a^1, \ldots, a^n)$ by choice. If $|K| > 1$ in step 3 then $\overline{a}^0 \in cone(\overline{a}^1, \ldots, \overline{a}^n)$ using the same arguments as those found in the proof of Theorem 1. If $|K| = 1$ then since $\overline{a}^0 \in cone(a^1, \ldots, a^n)$ and $\gamma^k \overline{a}^0 = 0$ it follows that $\overline{a}^0$ can be written as a non-negative combination of the set of vectors $\{a^i : i \in I_0^C\}$. Since $\{a^i : i \in I_0^C\} \subseteq \{\overline{a}^1, \ldots, \overline{a}^n\}$ by the operation of the algorithm in step 5, $\overline{a}^0 \in cone(\overline{a}^1, \ldots, \overline{a}^n)$.

Next, we show that when the algorithm terminates in step 3 no ascent direction exists and when it terminates in step 4 an ascent direction has been found. Using the same arguments as in the proof of Theorem 1 all constraints $a^i x \leq b_i$ encountered in the course of the algorithm satisfy $a^i \overline{x} = b_i$ and if the algorithm terminates in step 3 with $|K| > 1$ then no ascent direction exists. Since $a^0 \in cone(a^1, \ldots, a^n)$ at each iteration it follows by the definition of $\hat{\alpha}$ in step 2 that if $\hat{\alpha} \geq 1$ then $c \in cone(a^1, \ldots, a^n)$. Thus, $c = \omega_1 a^1 + \cdots + \omega_n a^n$ for some collection of $\omega_i \geq 0$ and so by Farkas' lemma no ascent direction in (P) exists at $\overline{x}$ upon termination in step 3. If $|K| > 1$ then using the same arguments as in the proof of Theorem 1 it follows that if the algorithm terminates in step 4 then d is an ascent direction. Thus, consider the case $|K| = 1$. By the choice of d and $\hat{\alpha}$ it is true that $d\overline{a}^0 = d(a^0 + \hat{\alpha}(c - a^0)) = 0$ and $d(a^0 + (\hat{\alpha} + \varepsilon)(c - a^0)) > 0$ for any $\varepsilon > 0$. Further, since the algorithm terminates in step 2 if $\hat{\alpha} \geq 1$ it must be that $\hat{\alpha} < 1$, and since $\overline{a}^0 = a^0 + \hat{\alpha}(c - a^0)$

it follows that $c - a^0 = (1 - \hat{\alpha})^{-1}(c - \overline{a}^0)$. Thus,

$$d(a^0 + (\hat{\alpha} + \varepsilon)(c - a^0)) > 0$$

$$\Longleftrightarrow \quad d(a^0 + \hat{\alpha}(c - a^0)) + \varepsilon d(c - a^0) > 0$$

$$\Longleftrightarrow \quad \varepsilon d(c - a^0) > 0$$

$$\Longleftrightarrow \quad (1 - \hat{\alpha})^{-1}\varepsilon d(c - \overline{a}^0) > 0$$

$$\Longleftrightarrow \quad dc > d\overline{a}^0 = 0$$

which implies that d is a direction of increasing objective function value. When the algorithm terminates in step 4 the value $\hat{\theta}$ returned by the oracle for solving problem STEP is nonzero, implying d is a feasible direction of movement for (P) at $\bar{x}$ and therefore a true direction of ascent.

We continue by demonstrating that a vector a^s with the properties assumed in step 5 exists. If $|K| > 1$ in step 3 then an a^s with the desired properties can be shown to exist using the same arguments as those found in the proof of Theorem 1, so consider the case $|K| = 1$. Step 5 of the algorithm is reached only if a value of $\hat{\theta} = 0$ is returned by the oracle for solving problem STEP, and by the definition of problem STEP it follows that $da^r > 0$. As the set of points $\{a^1, \ldots, a^n\} - \{a^s\}$ with s the unique element in K are linearly independent and satisfy $da^i = 0$, it follows that the points $\{a^1, \ldots, a^n\} \cup \{a^r\} - \{a^s\}$ are linearly independent as required.

We conclude by demonstrating that the algorithm is finite. To this end we first show that if $|K| = 1$ at some iteration then $\|c - a^0\|$ strictly decreases in the following iteration. To see this, let k be the unique index in K and let $\overline{a}^1, \ldots, \overline{a}^n$ be indexed so that $\overline{a}^k = a^r$ and $\overline{a}^i = a^i$ for $i > k$. Further, let $\overline{\gamma}^i$, $\overline{S}_i$, and $\overline{K}$ denote the values of γ^i, S_i, and K in the next iteration of the algorithm. When $|K| = 1$ the point $\overline{a}^0$ is in the strict relative

15

interior of the facet $cone(a^1, \ldots, a^{k-1}, a^{k+1}, \ldots, a^n)$ of $cone(a^1, \ldots, a^n)$ and is therefore in the strict relative interior of the facet $cone(\bar{a}^1, \ldots, \bar{a}^{k-1}, \bar{a}^{k+1}, \ldots, \bar{a}^n)$ of $cone(\bar{a}^1, \ldots, \bar{a}^n)$. Since $da^r = \gamma^k \bar{a}^k > 0$ it must be that $\bar{\gamma}^k = -\gamma^k$ in order for $\bar{\gamma}^k \bar{a}^i = 0$ for all $i \in \overline{S}_k$ and $\bar{\gamma}^k \bar{a}^k \leq 0$, and with $\bar{a}^0$ in the strict relative interior of this facet it follows that $\bar{a}^0 + \alpha w \in cone(\bar{a}^1, \ldots, \bar{a}^n)$ for any w satisfying $\bar{\gamma}^k w \leq 0$ and $\alpha > 0$ sufficiently small. We show that $\bar{\gamma}^k(c - \bar{a}^0) \leq 0$ as this will be sufficient to prove the strict decrease claim. Note that γ^k must satisfy $\gamma^k(c - a^0) > 0$, for if not then since $|K| = 1$, $\gamma^k(a^0 + \alpha(c - a^0)) \leq 0$ for all $\gamma^1, \ldots, \gamma^n$ for some $\alpha > \hat{\alpha}$, which contradicts the choice of $\hat{\alpha}$. Further, with $|K| = 1$ by assumption, $c - \bar{a}^0 = c - (a^0 + \hat{\alpha}(c - a^0)) = (1 - \hat{\alpha})(c - a^0)$. Thus, $\bar{\gamma}^k(c - \bar{a}^0) = \bar{\gamma}^k(1 - \hat{\alpha})(c - a^0) = -\gamma^k(1 - \hat{\alpha})(c - a^0) \leq 0$, implying that $\bar{a}^0 + \alpha(c - \bar{a}^0) \in cone(\bar{a}^1, \ldots, \bar{a}^n)$ for some sufficiently small $\alpha > 0$. If $|\overline{K}| = 1$ at the following iteration then the new value of a^0 is $\bar{a}^0 + \alpha(c - \bar{a}^0)$ with $\alpha > 0$, while if $|\overline{K}| > 1$ then the new value of a^0 is defined as the closest point to c in $cone(\bar{a}^1, \ldots, \bar{a}^n)$. In either case, since $\bar{a}^0 + \alpha(c - \bar{a}^0) \in cone(\bar{a}^1, \ldots, \bar{a}^n)$ for some $\alpha > 0$ it follows that $\|c - a^0\|$ strictly decreases at the following iteration as claimed.

Thus, consider a sequence of iterations in which $|K| = 1$ and let $\hat{a}$ be the initial value of a^0 in this sequence of iterations. At each iteration a^0 lies on the line segment connecting $\hat{a}$ and c, is strictly closer to c than in the previous iteration, and satisfies $\gamma a^0 = 0$ for some new constraint $\gamma a = 0$ uniquely defined by $n - 1$ points a^i that are rows of A. As there are a finite number of constraints $\gamma a = 0$ defined in this way, such a sequence of iterations must terminate either by finding a direction of ascent, proving that no such direction exists, or by entering an iteration in which $|K| > 1$. In an iteration in which $|K| > 1$, a^0 is redefined so that $\|c - a^0\| \leq \|c - a\|$ for all $a \in cone(a^1, \ldots, a^n)$ so that $\|c - a^0\|$ does not increase. Further, in the following iteration $\|c - a^0\|$ strictly decreases using arguments from the proof

of Theorem 1 in the case $|K| > 1$ and arguments similar to those made above in the case $|K| = 1$. Since $\|c - a^0\|$ never increases it is never possible for a^0 to again be contained in $cone(a^1, \ldots, a^n)$. As there are a finite number of cones defined by subsets of n rows of A, it follows that the algorithm is finite. $\quad\square$

It is of interest to compare the computational work required at each iteration of algorithm Close and algorithm Steep. When $|K| = 1$ in algorithm Close it can be seen that the algorithms are all but identical except in step 2, and the computational burden in both algorithms lies almost entirely in step 1. Indeed, for all practical purposes the algorithms are identical in the amount of work required per iteration if $|K| = 1$. As an aside, one intuitively pleasing aspect of closest ascent is that it provides an easily calculated measure of progress toward resolution of degeneracy, namely, the decreasing values of $\|c - a^0\|$, which steepest edge does not.

The question remains as to the likelihood that $|K| = 1$ in algorithm Close. From the definition of algorithm Close it can be seen that at any iteration of the algorithm the probability that $|K| \neq 1$ is the probability that when the ray $a^0 + \alpha(c - a^0)$ first intersects one of n planes $\gamma^k a \leq 0$ it simultaneously intersects at least one other plane. Over the course of the algorithm, the probability of finding $|K| \neq 1$ at some iteration is bounded above by the probability that the line segment connecting c and the initial a^0 chosen in step 0 never intersects more than one of a finite number of planes at a time. The finiteness of the number of planes under consideration follows from the fact that the planes encountered in algorithm Close are defined so as to contain the origin and $n - 1$ linearly independent rows of A. As the following theorem formalizes, the flexibility in the initial choice of a^0 guarantees that with probability 1 each iteration of the algorithm will encounter $|K| = 1$

independent of the given linear program.

Theorem 3 *Let $\gamma^i a = 0$, $i = 1, \ldots, N$, be a finite collection of planes in $\mathbb{R}^n$, let c be a distinguished point, and let a^0 be a point chosen at random from any full dimensional ball $B \subseteq \mathbb{R}^n$. The probability that any point a on the line segment connecting a^0 and c simultaneously satisfies two or more planes at equality is 0.*

Proof. Let U_{ij} denote the set of points determined by the intersection of the planes $\gamma^i a = 0$ and $\gamma^j a = 0$ (including the possibility of $\emptyset$) and let $\mathcal{U} = \bigcup U_{ij}$. If $c \in U_{ij}$ or $U_{ij} = \emptyset$ let $V_{ij} = U_{ij}$ and if $c \notin U_{ij}$ let V_{ij} denote the unique $n - 1$ dimensional affine set of points containing U_{ij} and c. Let $\mathcal{V} = \bigcup V_{ij}$. If the line segment connecting a^0 and c satisfies two or more planes at equality at some point a then since $a \in V_{ij}$ for some $V_{ij} \in \mathcal{V}$ and $c \in V_{ij}$ it must be that $a^0 \in V_{ij}$. Since $\mathcal{V}$ represents the set of points in a finite collection of affine sets of dimension $n - 1$ or less, the probability that a point chosen at random in any n dimensional ball B will lie in $\mathcal{V} \cap B$ is therefore 0, completing the proof. □

It cannot be overemphasized that the probability in the previous theorem is very different from most other probabilities arising in the discussion of degeneracy. For example, the probability of degeneracy occurring in a linear program is 0 in most natural probabilistic models for generating a random linear program, yet degeneracy is pervasive in most real linear programs. The probability calculated in the previous theorem, however, hypothesizes a *given* linear program and demonstrates that the freedom in choosing the initial value a^0 is what leads to the desired probability of 0.

The implication of the previous theorem is that in practice the need to find a^0 closest to c in $cone(a^1, \ldots, a^n)$ should occur very rarely, with a finite probability arising due to numerical

issues associated with finite precision arithmetic. In these instances, rather than find such a point a^0 it is preferable to find a new point a^0 closer to c through local adjustments to a^0. Properly developed, such an algorithm maintains finiteness while allowing for the extra freedom to locally adjust a^0.

Magnanti and Orlin (1988) observe that in recent probabilistic studies of objective function parametric linear programming algorithms "the probability distribution of a parametric linear program is chosen so that the problem almost always satisfies a condition [that ensures finite convergence of] the parametric algorithm...even if it is degenerate." From their work it appears that a stronger claim similar to the result found in Theorem 3 can be made. When solving a linear program using a parametric programming algorithm there is usually some flexibility in the choice of initial objective function. Choosing this objective function randomly from the set of potential choices should ensure that degeneracy is not encountered in the parametric algorithm independent of the underlying linear program.

6 Conclusions

Two procedures have been presented for resolving degeneracy in linear programs. The steepest ascent procedure, while not computationally attractive, shed some light on the practical efficiency of the steepest edge procedure commonly used in practice to solve large and highly degenerate linear programs. Ideas found in the steepest ascent procedure were then used to develop a closest ascent procedure that with probability 1 had nearly identical computational requirements per iteration as steepest edge but was guaranteed to finitely terminate. All of the procedures discussed had the unique advantage that the pivot element is chosen without knowledge of the set of all active constraints at a point of degeneracy.

A number of interesting questions remain. While the author suspects that there exist examples in which steepest edge leads to an infinite number of pivots, the question of whether or not steepest edge is a degeneracy resolution procedure remains unresolved. The relative number of iterations required in practice by steepest edge, steepest ascent, and closest ascent also remains an interesting question. Steepest ascent, while probably requiring the fewest iterations, is likely to be too expensive per iteration to warrant consideration in linear programming codes. However, the number of iterations required by closest ascent should be similar to the number required by steepest edge, in which case closest ascent should prove to be an effective practical procedure for degeneracy resolution.

7   Acknowledgements

References

[1] F. Barahona and H. Titan, "Max mean cuts and max cuts," IBM research report, Yorktown Heights (1990).

[2] R. E. Bixby, unpublished computational results (1991).

[3] R. E. Bixby and S. Saigal, "Optimizing over the cut cone: a new polyhedral algorithm for the maximum-weight cut problem," Doctoral thesis, Department of Mathematical Sciences, Rice University (1991).

[4] R. G. Bland, "New finite rules for the simplex method," *Mathematics of Operations Research* **2** (1977) 103-107.

[5] E. A. Boyd, "Fenchel cutting planes for linear integer programming problems," TR90-20, Department of Mathematical Sciences, Rice University (1990).

[6] S. C. Boyd and W. R. Pulleyblank, "Optimizing over the subtour polytope of the travelling salesman problem," *Mathematical Programming* **49** (1990/91) 163-187.

[7] A. Charnes, "Optimality and degeneracy in linear programming," *Econometrica* **20** (1952) 160-170.

[8] W. H. Cunningham, "Theoretical properties of the network simplex algorithm," *Mathematics of Operations Research* **4** (1979) 196-208.

[9] H. Crowder and J. M. Hattingh, "Partially normalized pivot selection in linear programming," IBM research report RC 4918, Yorktown Heights (1974).

[10] G. B. Dantzig, A. Orden, and P. Wolfe, "The generalized simplex method for minimizing a linear form under linear inequality restraints," *Pacific Journal of Mathematics* **5** (1955) 183-195.

[11] J. C. Dickson and F. P. Frederick, "A decision rule for improved efficiency in solving linear programming problems with the simplex method," *Communications of the ACM* **3** (1960) 509-512.

[12] J. Forrest and D. Goldfarb, "Steepest edge simplex algorithms for linear programming," IBM research report, Yorktown Heights (1991).

[13] D. Goldfarb and J. K. Reid, "A practicable steepest-edge simplex algorithm," *Mathematical Programming* **12** (1977) 361-371.

[14] P. M. J. Harris, "Pivot selection rules of the Devex LP code," *Mathematical Programming* **5** (1973) 1-28.

[15] K. L. Hoffman and M. Padberg, "Solving large set partitioning problems arising in the airline industry," TIMS/ORSA meeting presentation, Nashville (1991).

[16] H. W. Kuhn and R. E. Quandt, "An experimental study of the simplex algorithm," in Metropolis *et. al., eds., Proceedings of symposia in applied mathematics*, vol. XV (American Mathematical Society, Providence, RI, 1963).

[17] T. L. Magnanti and J. B. Orlin, "Parametric linear programming and anti-cycling pivoting rules," *Mathematical Programming* **41** (1988) 317-325.

[18] D. M. Ryan and M. R. Osborne, "On the solution of highly degenerate linear programs," *Mathematical Programming* **41** (1988) 385-392.

[19] P. Wolfe, "A Technique for resolving degeneracy in linear programming," *Journal of SIAM* **11** (1963) 205-211.

[20] P. Wolfe and L. Cutler, "Experiments in linear programming," in Graves and Wolfe, *eds., Recent advances in mathematical programming* (McGraw-Hill, New York, 1963).